Pixel.js and the Classification Method (CM) as a Rodan Workflow

Eric Liu and Martha Thomae

DDMAL Lab, McGill University
SIMSSA Workshop XIV
28 May, 2018
Rodan

- Workflow engine
- Developed primarily by Andrew Hankinson, Ryan Bannon, and Ling Xiao Yang
- Users upload files (resources)
- Allows chaining of different processes (jobs) to create a workflow

Objectives

● Classify large manuscripts into layers using Pixel.js
  ○ Manuscript: notes
  ○ Map: bodies of water, parks

● Implement a Rodan workflow with Pixel and CM
  ○ Generate a background layer in Pixel.js
The Classification Method with Pixel.js

- Developed by Jorge Calvo Zaragoza

- Workflow
  - Trainer: input layers, output model
  - Classifier: input model, output classified image

- Classifying shortcut
  - Saves time and user-labor

Pixel.js

- Developed by Zeyad Saleh & Ké Zhang
- A web based drawing/layering plugin for ground truth creation
- Built on Diva.js
  - Web based document viewer
  - Andrew Hankinson, Wendy Liu, Laurent Pugin, Ichiro Fujinaga

How does it work?

- Divide images into color coded layers
  - Pixel tools: brush, rectangle, eraser, select
  - Layer for music symbols, staff lines, text, background
Background Layer in Pixel.js

- The CM trainer requires all layers
  - No concept of “everything else”

- Original implementation
  - Path/shape wise generation
  - Not compatible with the CM classifier, some bugs
Current Development

- Pixel-wise generation
  - Fixes bugs
  - Slower (minutes)

- Button, progress bar
Rodan Workflow

Pixel.js: user classifies a portion of a score

Trainer: generates a training model from ground truth

Classifier: use model to classify image

Pixel.js: correct classification

Repeat: optimize training model
Demo
Thank you!